

THE STATA JOURNAL

adata, citation and similar papers at core.ac.uk

brought to you by

provided by Research Papers in

H. Joseph Newton
Department of Statistics
Texas A & M University
College Station, Texas 77843
979-845-3142; FAX 979-845-3144
jnewton@stata-journal.com

Nicholas J. Cox
Department of Geography
University of Durham
South Road
Durham City DH1 3LE UK
n.j.cox@stata-journal.com

Associate Editors

Christopher Baum
Boston College
Rino Bellocco
Karolinska Institutet
David Clayton
Cambridge Inst. for Medical Research
Mario A. Cleves
Univ. of Arkansas for Medical Sciences
Charles Franklin
University of Wisconsin, Madison
Joanne M. Garrett
University of North Carolina
Allan Gregory
Queen's University
James Hardin
University of South Carolina
Stephen Jenkins
University of Essex
Jens Lauritsen
Odense University Hospital
Stanley Lemeshow
Ohio State University

J. Scott Long
Indiana University
Thomas Lumley
University of Washington, Seattle
Roger Newson
King's College, London
Marcello Pagano
Harvard School of Public Health
Sophia Rabe-Hesketh
University of California, Berkeley
J. Patrick Royston
MRC Clinical Trials Unit, London
Philip Ryan
University of Adelaide
Mark E. Schaffer
Heriot-Watt University, Edinburgh
Jeroen Weesie
Utrecht University
Jeffrey Wooldridge
Michigan State University

Stata Press Production Manager

Lisa Gilmore

Copyright Statement: The Stata Journal and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the Stata Journal.

The articles appearing in the Stata Journal may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the Stata Journal.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the Stata Journal, in whole or in part, on publicly accessible web sites, file servers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the Stata Journal or the supporting files understand that such use is made without warranty of any kind, by either the Stata Journal, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the Stata Journal is to promote free communication among Stata users.

The *Stata Technical Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press, and Stata is a registered trademark of StataCorp LP.

Stata tip 6: Inserting awkward characters in the plot

Nicholas J. Cox, University of Durham, UK
n.j.cox@durham.ac.uk

Did you know about the function `char()`? `char(n)` returns the character corresponding to ASCII code *n* for $1 \leq n \leq 255$. There are several numbering schemes for so-called ASCII characters. Stata uses the ANSI scheme; a web search for “ANSI character set” will produce tables showing available characters. This may sound like an arcane programmer’s tool, but it offers a way to use awkward text characters—either those not available through your keyboard or those otherwise problematic in Stata. A key proviso, however, is that you must have such characters available in the font that you intend to use. Fonts available tend to vary not only with platform but even down to what is installed on your own system. Some good fonts for graphics, in particular, are Arial and Times New Roman.

Let us see how this works by considering the problem of inserting awkward characters in your Stata graphs, say as part of some plot or axis title. Some examples of possibly useful characters are

<code>char(133)</code>	ellipsis	...
<code>char(134)</code>	dagger	†
<code>char(135)</code>	double dagger	‡
<code>char(169)</code>	copyright	©
<code>char(176)</code>	degree symbol	°
<code>char(177)</code>	plus or minus	±
<code>char(178)</code>	superscript 2	²
<code>char(179)</code>	superscript 3	³
<code>char(181)</code>	micro symbol	μ
<code>char(188)</code>	one-fourth	$\frac{1}{4}$
<code>char(189)</code>	one-half	$\frac{1}{2}$
<code>char(190)</code>	three-fourths	$\frac{3}{4}$
<code>char(215)</code>	multiply	×

There are many others that might be useful to you, including a large selection of accented letters of the alphabet. You can use such characters indirectly or directly. The indirect way is to place such characters in a local macro and then to refer to that macro within the same program or do-file.

For example, I use data on river discharge, for which the standard units are cubic meters per second. I can get the cube power in an axis title like this:

```
. local cube = char(179)
. scatter whatever, xtitle("discharge, mcube/s")
```

Or, I have used Hanning, a binomial filter of length 3:

```
. local half = char(189)
. local quarter = char(188)
. twoway connected whatever,
>       title("Smoothing with weights 'quarter':'half':'quarter'")
```

The direct way is to get a macro evaluation on the fly. You can write `'=char(176)'` and, in one step, get the degree symbol (for temperatures or compass bearings). This feature was introduced in Stata 7 but not documented until Stata 8. See [P] **macro**.

`char()` has many other uses besides graphs. Suppose that a string variable contains fields separated by tabs. For example, `insheet` leaves tabs unchanged. Knowing that a tab is `char(9)`, we can

```
. split data, p('=char(9)') destring
```

Note that `p(char(9))` would not work. The argument to the `parse()` option is taken literally, but the function is evaluated on the fly as part of macro substitution.

Note also that the SMCL directive `{c #}` may be used for some but not all of these purposes. See [P] **smcl**. Thus,

```
. scatter whatever, xtitle("discharge, m{c 179}/s")
```

would work, but using a SMCL directive would not work as desired with `split`.